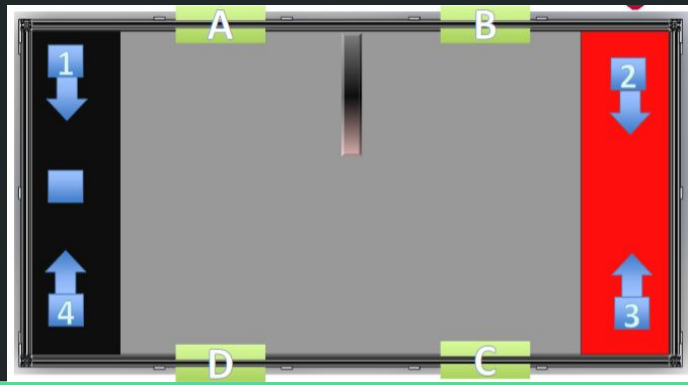
E-121 Oral Presentation Group 5

Ben Gilmore, Evan Eleftheriadis, Jose de la Cruz

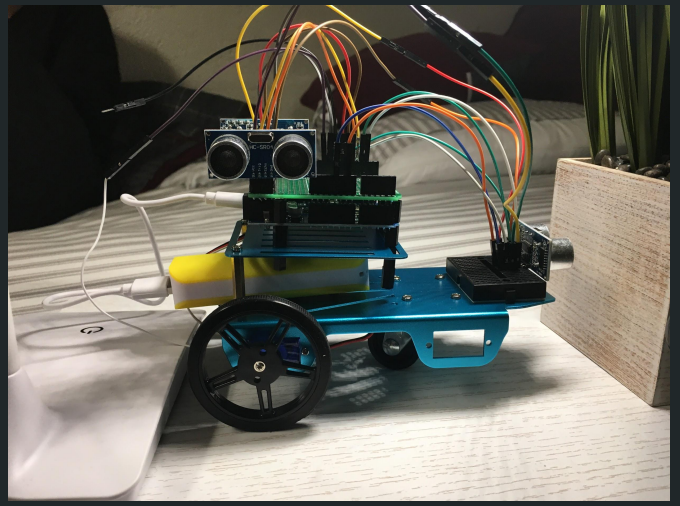
Tests

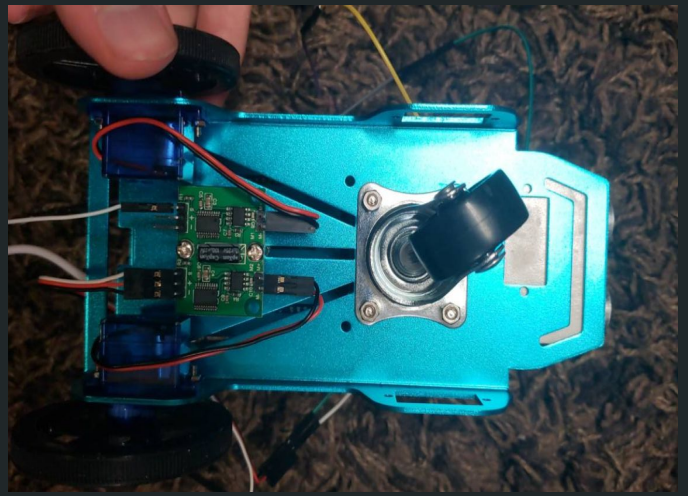
- In the first road test the robot had to start at one end of the track and move in a straight line and stop within two inches from the wall
- The second test was the left handed loop which required to go around an inner wall of the track three times, as well as do it quickly
- The final test will have the robot start in one of four locations, it will then have to hit the targets in the order BDAC, the robot will then have to return to its starting position



Parts

- 3 sensors: one on the front and two on the sides
  - 1 originally for 1st trial test, then as we progressed we used one in front and 2 on the sides on the sautered board
- 3 wheels: two attached to the motors on the sides and one free spinning wheel attached to the bottom
- 2 chasii: one for robot body and one to secure the arduino board and sautered board



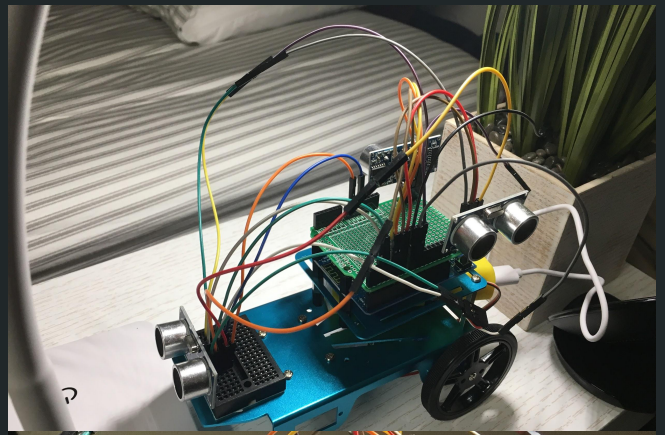


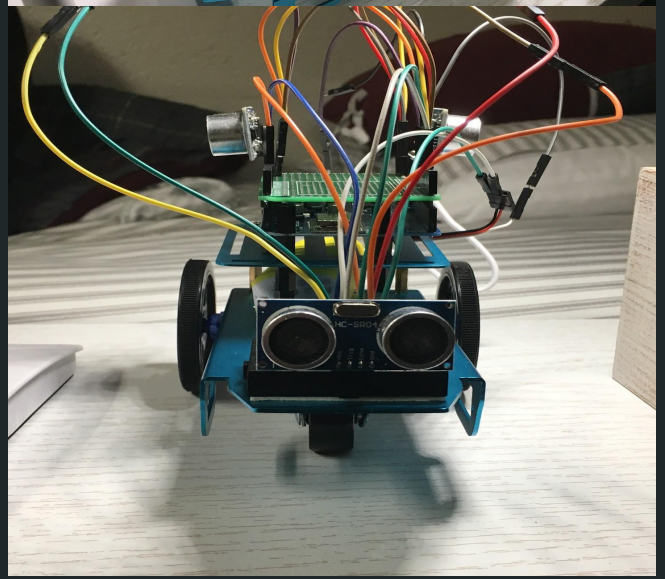
Assembly

- The first step was attaching the wheel and the board on the bottom as well as the two motors on the side and the two wheels.
- The next step was to put the the bracket on the supports so that we could put the board on and attach the sensors
- We have the breadboard at the front of the robot with the third sensor
- The team had to assemble the robot twice

Sensor Placement

- There are two sensors on the bracket towards the back of the robot which were used in the left hand loop test. These are right above the board and the [insert part about soldering]
- The other sensor is located at the front of the robot. This will be used to judge the distance in front of the robot. It is on the breadboard that is attached to the arduino board





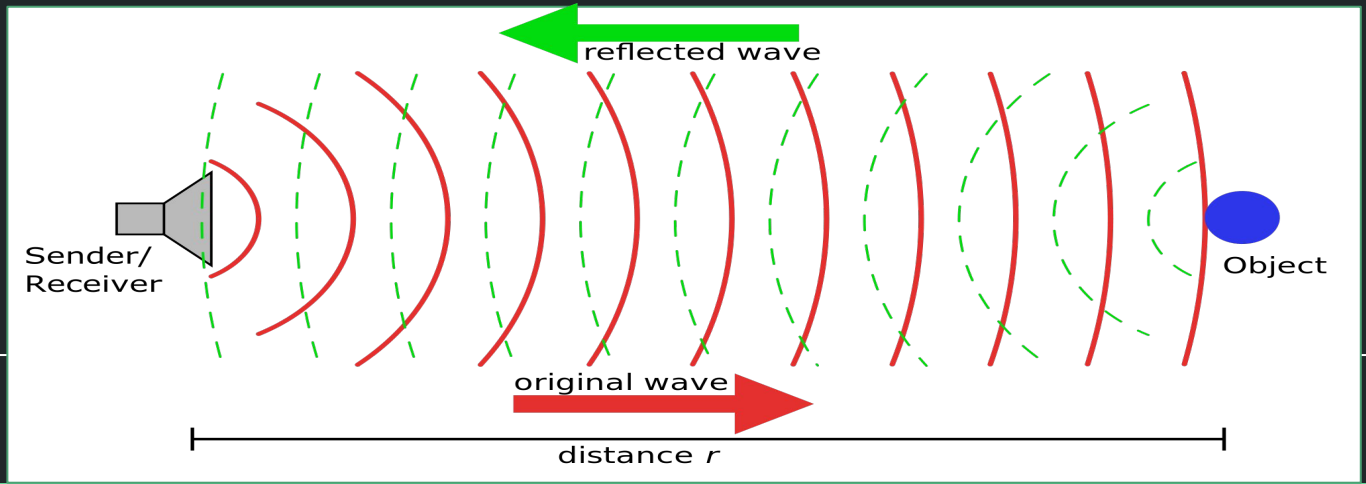
Motor Mechanics

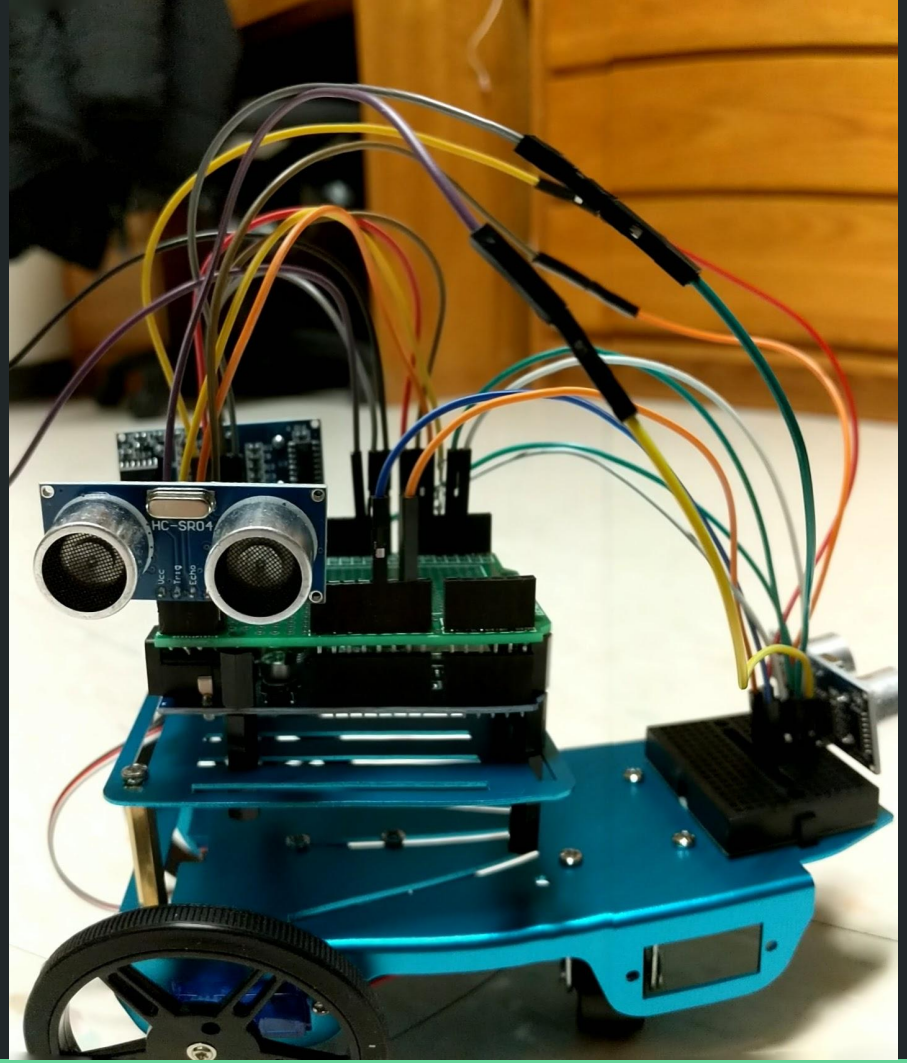
- Due to the fact the motors are placed in opposite directions, one code must have the first wheel spinning forwards and the other spinning in reverse
- For the coding, force of friction and error within the motors must be taken into account when inputting the values into the code for the motor's acceleration/ deceleration

0	Full speed
90	Stop
180	Full speed opposite direction

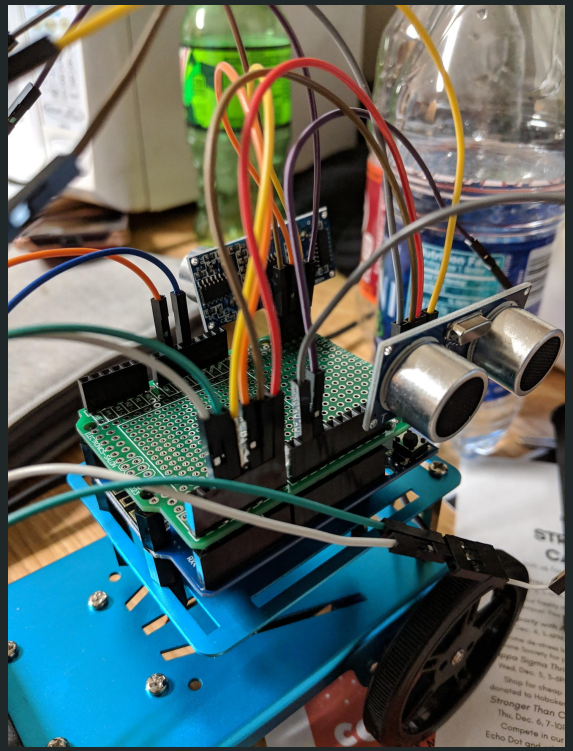
Sensor Mechanics

- The sensor works like echolocation or radar
- The trigger pin sends a signal outwards
- The echo pin relays the signal as it bounces back to the robot
- Distance is returned as a integer that can be used to measure distance




HC-SR04
Vcc
Trig
Echo

Pin Layout



Pin	Function
2	Front Sensor Output
3	Front Sensor Input
4	Right Sensor Output
5	Right Sensor Input
6	Left Sensor Output
7	Left Sensor Input
9	Left Motor
10	Right Motor

Programing

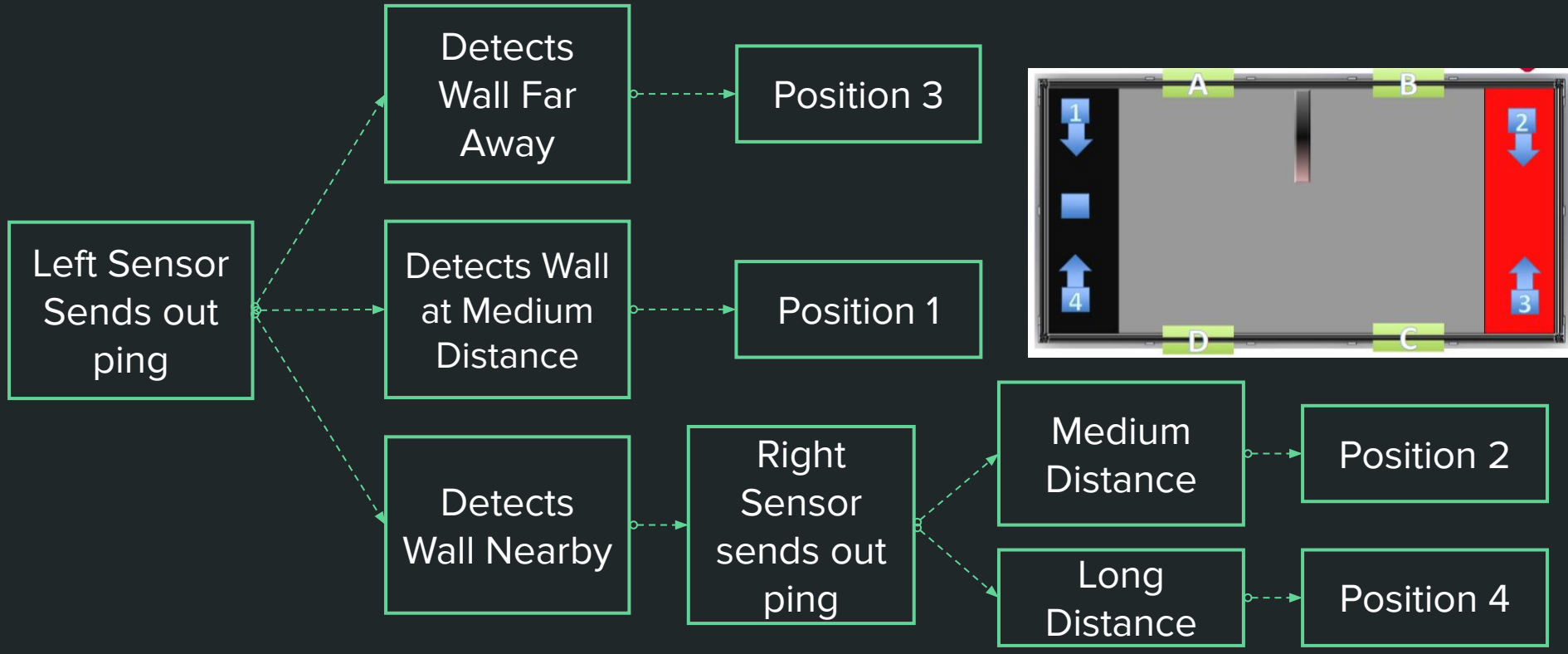
- Most of the code is pretty minimalist
- Basic use of variables
- No creation of functions

```
void loop() {

Serial.println(ultrasonicPing(D11,D12));
delay(10);
long distance = ultrasonicPing(D11,D12); //distance between sensor and wall

if (distance>500 && distance <750){ //straight
motor1.write(0);
motor2.write(180);
}
else if (distance>800){ //left
motor1.write(0);
motor2.write(100);
}
else{ //right
motor1.write(85);
motor2.write(180);
}
```

Programing flow-chart (Go argue in some other slide)



Errors/ Discrepancies

- One of the sensors provided in the robot kit was faulty and didn’t perform the expected action which caused us to dedicate time in finding the error in our code which was correct from the start
- Sensor would stop in time before a cardboard wall yet didn’t recognize the glass surface in the nascar loop and continuously crashed into the wall
- Wheels may lay at a angle and cause unwanted degree changes in the robot’s direction
- Wheel without motor in the front had to be aligned straight as to not throw the robot off its course
- Lost robot made us lose a week of fine tuning the assembly but allowed us to work on the code and the technical report